DBS EXTERNAL API GATEWAY

CUSTOMER ONBOARDING GUIDE

Version 2.1

DBS makes no warranties or representations with respect to the content hereof and specifically disclaim any implied warranties of merchantability or fitness for any particular purpose. DBS assumes no responsibility for any errors that may appear in this document. The information contained in this document is subject to change without notice. DBS reserves the right to make any such changes without obligation to notify any person of such revision or changes. DBS makes no commitment to keep the information contained herein up to date.

Document Sign-Off

Revision History

Revision Date	Version	Summary of Changes
14/10/2019	2.0	-Added Revision History for tracking purpose. -Added Sync and Async API exceptional handling.
31/10/2019	2.1	-Added Figure 3 of PGP encryption flow for Outbound API

Table of Contents

Technical Architecture of DBS EXTERNAL API GATEWAY

Figure 1 details the system architecture of DBS EXTERNAL API GATEWAY system. The network connection to DBS will be over the Internet.

Encryption software must be installed at Customer's Backend Application Server to perform the necessary encryption and signing of files.

Security Features of DBS EXTERNAL API GATEWAY includes:

- Secured communication channel using SSL
- Confidentiality, integrity and authenticity of the message using PGP or JWT encryption
- Digitally signed using PGP or JWT
- Acknowledgement of message received through HTTPS JSON response

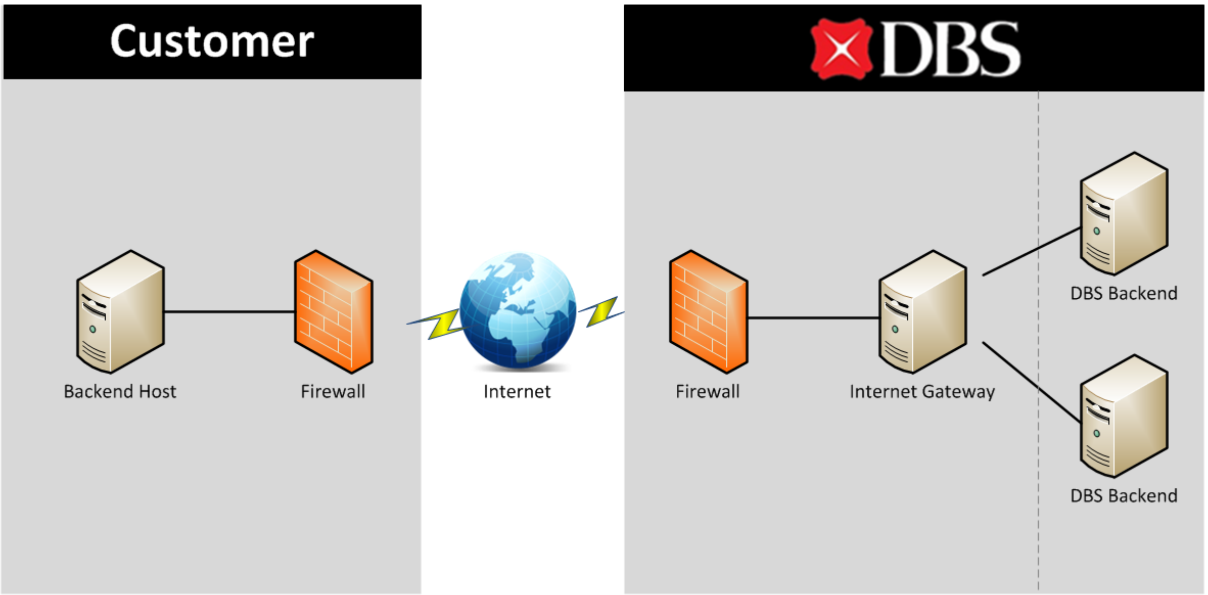


Figure 1 : DBS EXTERNAL API GATEWAY Network Architecture

1 Introduction

This section documents the purpose, background and scope of this DBS EXTERNAL API GATEWAY Onboarding Guide. The target audiences are DBS EXTERNAL API GATEWAY customer’s technical team.

1.1 Purpose

This guide provides an overall picture on DBS EXTERNAL API GATEWAY network architecture and specifies the onboarding activities required by customers who would like to subscribe to DBS EXTERNAL API GATEWAY API services.

1.2 Background

The primary business objective of DBS EXTERNAL API GATEWAY is to facilitate customers to be able to connect to DBS services via online APIs. This provides an additional digital channel for customers such as online sites and payment providers, while improving the service quality delivered to DBS EXTERNAL API GATEWAY customers.

1.3 Scope

The guide details the mandatory setup activities and information to be exchanged between DBS and DBS EXTERNAL API GATEWAY customer during the onboarding process. The list of items to be discussed in this guide includes:

1. Technical architecture of DBS EXTERNAL API GATEWAY
2. Message transfer format
3. Encryption and decryption
4. Information to be exchanged between DBS and customers
5. Onboarding activities to be carried out by DBS and customers

1.4 Requirements

This section describes the requirements for onboarding to DBS EXTERNAL API GATEWAY.

<u>Network</u>

- Internet connection to DBS EXTERNAL API GATEWAY
- Firewall rules to permit network traffic between DBS and DBS EXTERNAL API GATEWAY customer

Encryption Software

Pretty Good Privacy (PGP) encryption is used for the encryption/decryption and signing/verification of the message.

Alternatively, JSON Web (JW) Encryption and Signing method can also be used for the encryption/decryption and signing/verification of the message.

Refer to section 1.6 for more information on encryption and decryption, and signing and verification.

Each party is responsible to ensure that the preparatory setup (e.g. hardware, software, network, firewall, information exchange etc.) is completed before testing or implementation in production.

1.5 Message Transfer Format

The DBS EXTERNAL API GATEWAY provides a collection of REST APIs that can be called. An API can be called via a HTTPS request. The message format used by DBS EXTERNAL API GATEWAY is in JSON format.

A message typically consists of 3 parts – the header, body(payload) and signature.

The header is used to identify the type of message and to route it to the correct API.

The payload is then Encrypted and Digitally signed using PGP mechanism, while if using the JWE (JSON Web Encryption) and JWS (JSON Web Signature) mechanism then payload is digitally signed and then encrypted.

PGP Message	JW Message	Description
Header	Header	HTTP Headers
Body (Encrypted using receiver's public key)	Body (Signed using sender's private key)	Plain JSON Payload
Signature (Encrypted payload is signed using sender's private key)	Encryption (Encrypt signed payload using receiver's public key)	Digitally Signed and Encrypted JSON Payload

Table 1: Sample message Overview

The sample message will be provided as part of API specification document.

1.6 Message Encryption

The following encryption software are supported by DBS EXTERNAL API GATEWAY:

1. Pretty Good Privacy (PGP)

- Supports up to RSA 2048 bit and AES encryption.
- Complies with the PKCS#7 standard
- The recommended key pair algorithm will be RSA 2048 bit
- For more information, please refer to www.pgp.com

Symmetric Key Encryption and Signature Algorithm

Customer recommended to comply to below encryption algorithm when encrypting and signing the payload.

1. Payload to be encrypted and then signed in **One Pass** method.

2. Compression Algorithm – **ZIP**

3. Symmetric Key Algorithm – **AES256**

4. Hash Algorithm - **SHA256**

5. Signing method – **Compressed**

Bank standard Allowed Algorithms

Hash Algorithm – SHA256, SHA384, SHA512

Symmetric Key Algorithm - AES128, AES192, AES256 and TRIPLE_DES(3DES)

Two pairs of private/public PGP keys are involved - one pair for Encrypt/Decrypt and another pair for Sign/Verify.

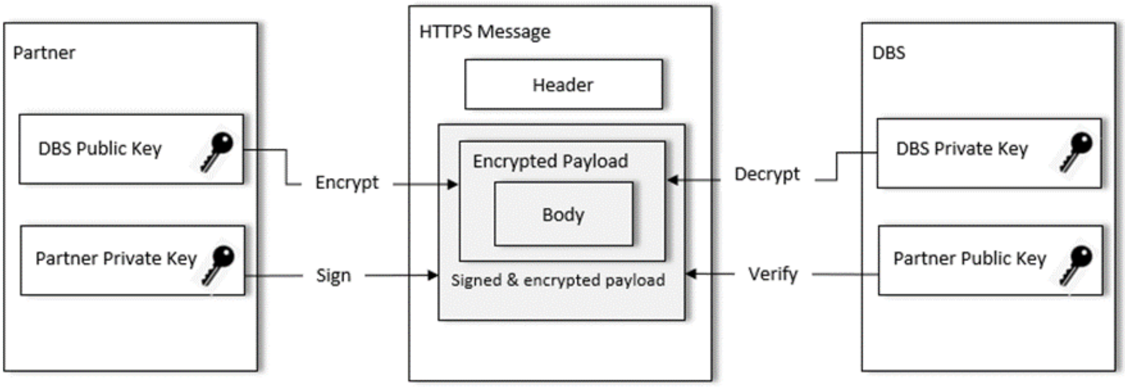


Figure 2: PGP Message Encryption Flow for Inbound API

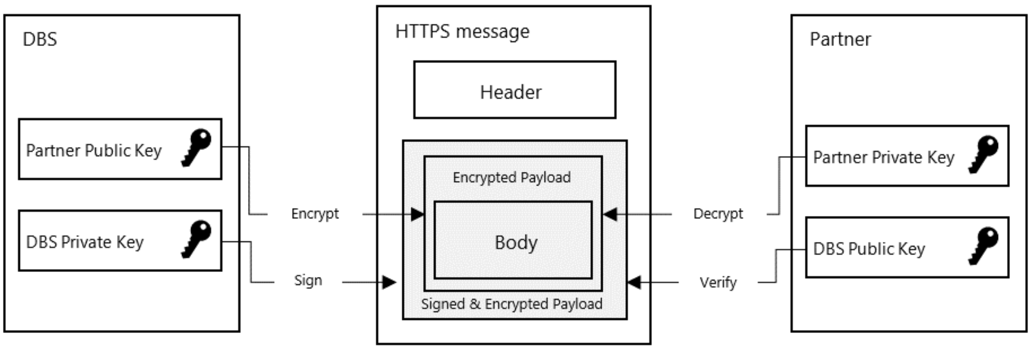


Figure 3: PGP Message Encryption Flow for Outbound APIs (e.g. ICN/CCA)

<u>Generation of PGP key pairs</u>

GNU Privacy Guard (GnuPG) tool can be used for generating and managing the PGP keys. Note that keys should be generated without a passphrase.

http://www.gnupg.org/

Example commands:

- Create a PGP key:

  ```
  gpg --gen-key
  ```

- View the PGP key:

  ```
  gpg -a --export
  ```

- Exports a public key to a file:

  ```
  gpg --export -u 'UserName ' -a -o public.key
  ```

- Exports a private key to a file:

  ```
  gpg --export-secret-keys -u 'UserName ' -a -o private.key
  ```

- Lists the private keys:

  ```
  gpg --list-secret-keys
  ```

2. <u>JWT (JSON Web Token)</u>

- The payload should be signed then encrypted.
- RSA key length should be at least 2048 bits.
- The Certificate should be in **X.509 certificate** and the extension can be .pem, .cer or .crt format.

Symmetric Key Encryption and Signature Algorithm

Customer needs to comply to below encryption algorithm when encrypting and signing the payload.

1. Signing algorithm - **RSASSA-PKCS1-v1_5 using SHA-256(RS256)**
2. Symmetric-key algorithm to use to encrypt the data - **AES_128_CBC_HMAC_SHA_256**

   Asymmetric algorithm used to encrypt symmetric key - **RSAES-PKCS1-V1_5 (RSA1_5)**

   Two pairs of private / public certificates () are involved - one pair for Encrypt/Decrypt and another pair for Sign/Verify.

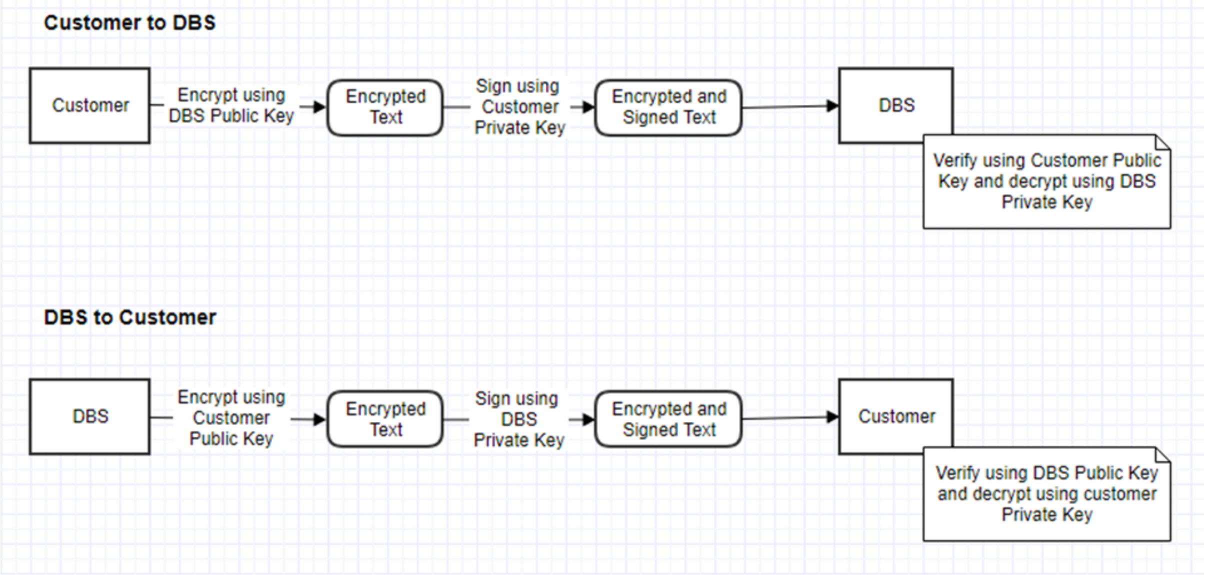

Customer to DBS
Customer
Encrypt using DBS Public Key
Encrypted Text
Sign using Customer Private Key
Encrypted and Signed Text
DBS
Verify using Customer Public Key and decrypt using DBS Private Key
DBS to Customer
DBS
Encrypt using Customer Public Key
Encrypted Text
Sign using DBS Private Key
Encrypted and Signed Text
Customer
Verify using DBS Public Key and decrypt using customer Private Key

Figure 4: JWT Message Encryption Flow

Encryption / Decryption JWT

Encryption: Public Certificate will be shared with the customer who is encrypting the payload using JWT.

Decryption: DBS will hold the private key for decryption of the payload.

orgId needs to be specified in the request header. Example: ORG_ID = SGSP02.

Sign / Verify JWT

Sign: Customer will hold the private key and will use it to sign the payload.

Verify: Public certificate will be shared with DBS for verification of the payload.

After decryption and verification is successful, the decrypted message payload will be sent to the back-end systems in DBS for processing. If there is any failure, an appropriate security related error message will be returned to the customer.

3. JWE (JSON Web Encryption) and JWS (JSON Web Signature)

- The payload should be signed then encrypted.

- RSA key length should be at least 2048 bits.

Symmetric Key Encryption and Signature Algorithm

Customer needs to comply to below encryption algorithm when encrypting and signing the payload.

3. Signing algorithm - **RSASSA-PKCS1-v1_5 using SHA-256(RS256)**
4. Symmetric-key algorithm to use to encrypt the data - **AES_128_CBC_HMAC_SHA_256**
5. Asymmetric algorithm used to encrypt symmetric key - **RSAES-PKCS1-V1_5 (RSA1_5)**

Two pairs of private / public keys () are involved - one pair for Encrypt/Decrypt and another pair for Sign/Verify.

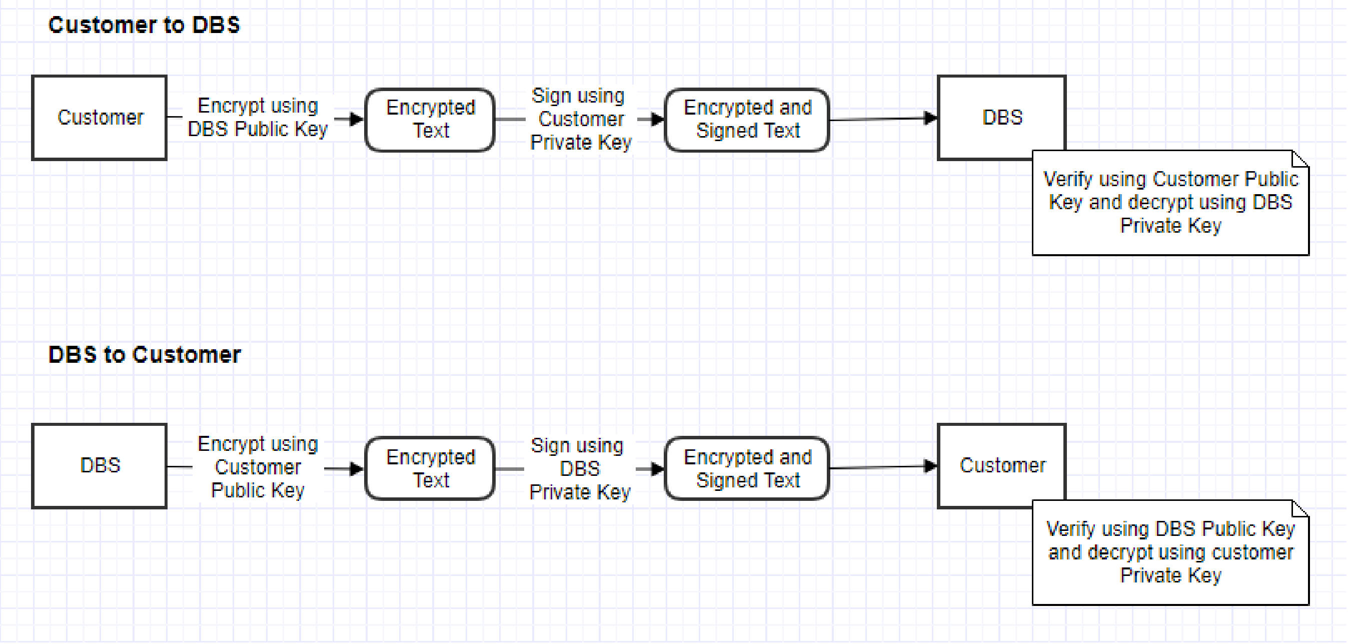

Customer to DBS
Customer
Encrypt using DBS Public Key
Encrypted Text
Sign using Customer Private Key
Encrypted and Signed Text
DBS
Verify using Customer Public Key and decrypt using DBS Private Key
DBS to Customer
DBS
Encrypt using Customer Public Key
Encrypted Text
Sign using DBS Private Key
Encrypted and Signed Text
Customer
Verify using DBS Public Key and decrypt using customer Private Key

Figure 5: JWT Message Encryption Flow

Encryption / Decryption

Encryption: Public key will be shared with the customer who is encrypting the payload using PGP.

Decryption: DBS will hold the private key for decryption of the payload.

orgId needs to be specified in the request header. Example: ORG_ID = SGSP02.

Sign / Verify

Sign: Customer will hold the private key and will use it to sign the payload.

Verify: Public key will be shared with DBS for verification of the payload.

After decryption and verification is successful, the decrypted message payload will be sent to the back-end systems in DBS for processing. If there is any failure, an appropriate security related error message will be returned back to the customer.

1.7 DBS Transport Layer Security standards

- SSL connection TLS 1.2 should be implemented for TLS support others are not supported.
- A Cipher Suite is a suite of cryptographic algorithms used by an SSL connection. Ciphers associate with RSA, AES128, AES256, SHA256 and SHA384 are acceptable.

Standard Cipher Suites that can be used for SSL connection (TLS1.2)

TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256

TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256

TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384TLS_RSA_WITH_AES_128_CBC_SHA256

TLS_RSA_WITH_AES_256_CBC_SHA256

1.8 Data Retention

All messages transferred through DBS EXTERNAL API GATEWAY will be kept online by backend systems for 90 days before they are archived to offline tapes. All the financial data will be kept up to 7 years on tape based on regulatory requirements.

1.9 Email and SMS Notification

DBS provides an email and SMS notification service to notify customers on the success of certain high value transactions, such as payments.

Depending on the use case, the notification can either be sent to the customer or the customer’s customers. For example, in the case of payments initiation, a notification can be send to the initiator (customer’s customers) upon a successful transfer of funds.

2 Information Exchange

Before the customer can perform testing with DBS, certain information and keys will need to be exchanged with DBS.

The following items need to be exchanged between the customer and DBS.

Environment	Production	UAT	Sandbox
Incoming to DBS			
Source Public IP	<Customer to provide>	<Customer to provide>	<Customer to provide>
Destination Public IP	103.4.36.89/103.4.38.89 (aping-ideal.dbs.com) Note:Whitelisting required for both the IP's	115.42.197.133 (aping-ideal-uat.dbs.com)	203.126.136.155 (api-ideal-staging.dbs.com)
Destination Port	443	443	443
CA Signed SSL cert	<DBS to provide>	<DBS to provide>	<DBS to provide>
Outgoing from DBS			
Source Public IP	203.127.89.206/32 203.116.36.92/32 128.106.20.192/26 203.116.37.64/26	203.127.89.206/32 203.116.36.92/32 128.106.20.192/26 203.116.37.64/26	203.127.89.206/32 203.116.36.92/32 128.106.20.192/26 203.116.37.64/26
Customer URL	<Customer to provide>	<Customer to provide>	<Customer to provide>
Customer Port	443	443	443
CA Signed SSL cert	<Customer to provide>	<Customer to provide>	<Customer to provide>
PGP/JWT Key Exchange			
Encryption Public Key	<DBS to provide>	<DBS to provide>	<DBS to provide>
Signature Verification Public Key	<Customer to provide>	<Customer to provide>	<Customer to provide>

Table 3: Information Exchange

3 On-Boarding Activities

The following subsection documents the activities and testing required at DBS and customer.

3.1 List of Activities

Before the interface testing could be performed, the following activities will need to be carried out:

No	Activity Item	Action Party	Duration	Remarks
1	Onboarding of profile			Items required (Sandbox, UAT& Production): ▪ API subscriptions ▪ Profile entitlements
2	Network Information Exchange	DBS & Customer	1 day	Items required (Sandbox, UAT& Production): ▪ IP Addresses ▪ Port number
3	Raise firewall change request for UAT & Production	DBS & Customer (where applicable)	1 day	2 weeks lead time to raise necessary change request for firewall rules change
4	Key Exchange	DBS & Customer	3 days	Items required (Sandbox, UAT & Production): ▪ PGP public and private keys
5	UAT Environment Setup	DBS	5 days	
6	Connectivity Test on UAT	DBS & Customer	2 days	Message transmission for both request and return. Encryption/decryption and sign/verify must work for both parties.
7	UAT transaction testing	Customer	5 days	API testing
8	UAT sign-off	Customer	1 day	
9	Raise change request for production deployment	DBS & Customer (where applicable)	1 day	2 weeks lead time to raise necessary change request for production deployment
10	Production deployment	DBS & Customer (where applicable)	1 day	
11	Connectivity test on Production	DBS & Customer	1 day	Message transmission for both request and return. Encryption/decryption and sign/verify must work for both parties.
12	Live Verification on Production	DBS & Customer	3 days	Customer to send a live transaction message. DBS to monitor end-to-end.

13	Post Implementation	DBS & Customer	3 days	Review of the entire implementation for feedback.

Table 4: List of activities for DBS and customer for Onboarding

Note: Estimated man-days may vary depending of the scope and complexity of the file transfer requirements and customer's environment.

3.2 Testing Requirements

In order to onboard to DBS EXTERNAL API GATEWAY, customers would need to perform testing with DBS before implementing in production. The testing includes both Connectivity Testing and User Acceptance Testing.

3.2.1 Connectivity Testing

The objective of Connectivity Testing includes the following:

- Telnet
- PGP Encryption/Decryption
- PGP Sign/Verify
- Message transmission

3.2.2 Sandbox Testing (Optional)

The objective of Sandbox includes the following:

- Sandbox Test environment to do preliminary API testing using Mock data.
- Message format
- Response message – message format & rejection codes

3.2.3 User Acceptance Testing (UAT)

The objective of UAT includes the following:

- Routing of the message based on the header to the relevant API
- Message format
- Response message – message format & rejection codes

Testing scope varies based on the API that the customer intends to subscribe to.

4 Appendix A: Additional API Specific Requirements

This section describes the additional information required by specific API.

4.1 Retail Customer

No	Client Onboarding Parameters	Values
1	client_id	DBS ORG-ID at DBS EXTERNAL API GATEWAY which would be assigned to corporate customer at the entity level
2	client_secret	<DBS to provide>
3	redirect_uri	<Customer to provide>
4	scope	ddaSetup, retrieveAccounts
5	Auth Code Expiration	3 mins
6	Access Token Expiration	15 mins

Table 5: List of activities for OAuth Onboarding for Retail Customers

4.2 Ideal Customer (ERP Integration)

No	Client Onboarding Parameters	Values
1	client_id	DBS ORG-ID at DBS EXTERNAL API GATEWAY which would be assigned to corporate customer at the entity level
2	client_secret	<DBS to provide>
3	redirect_uri	<Customer to provide>
4	Auth Code Expiration	30 seconds
5	Refresh Token Expiration	90 days
6	Access Token Expiration	4 hours
7	Cancel/Error Screen /Back button URL	<Customer to provide>
8	Learn More URL	<Customer to provide>
9	Contact Us URL	<Customer to provide>
10	Logo	<Customer to provide>

Table 6: List of activities for OAuth Onboarding for Ideal Customers

Below are some exceptional scenarios to be taken care of with action required from partner side:

Scenario	Expected Outcome	Error Codes	Action by Partner
Locked IDEAL Account	User can get statements after account is unlocked.	Allow token refresh so that user can resume when account is unlocked	No Action

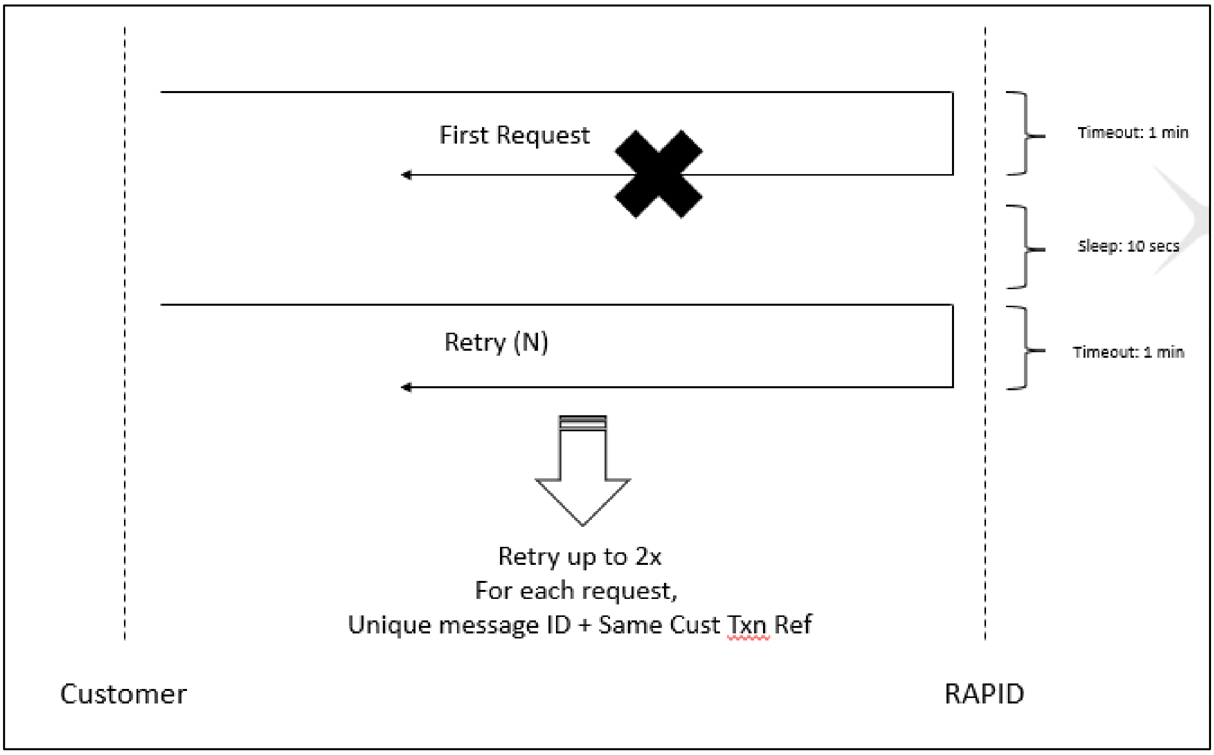
Deleted IDEAL account	User is unable to get statements.	Error code is E423	Remove token/access upon receiving this error code
Late generation of statement	User can get statements	Error code is MT940003	Perform retry upon receiving this error code
User un-provision statement on IDEAL	User should be informed to perform provision as he\she un-provisioned in Ideal.	Error code is E422	Partner to inform customers to provision upon receiving this error code
SGD accounts not available on Monday	User is unable to get statements.	Error code is IG940003	No Action
Organization id passed is incorrect or organization id in http header does not match payload org id	Unable to do transaction	A001	Verify and pass correct Org id
Maximum transaction transmission is exceeded	User can do transaction successfully	A002	Perform retry upon receiving this error code
Invalid Request	Unable to do transaction	A003	Verify the JSON format of payload and resend
Key pair used for encryption and signing are incorrect	Unable to decrypt and verify request	A004	Verify keys and mechanism & algorithms for encrypt and sign service. Refer section 1.6 for more info.
DBS External API Gateway not able to connect to backend.	User can do transaction successfully	A005	Perform retry after some time upon receiving this error code, if received again contact DBS support.

Table 7: Exception Scenarios

Below are **sync** API and **async** API exceptional handling diagram:

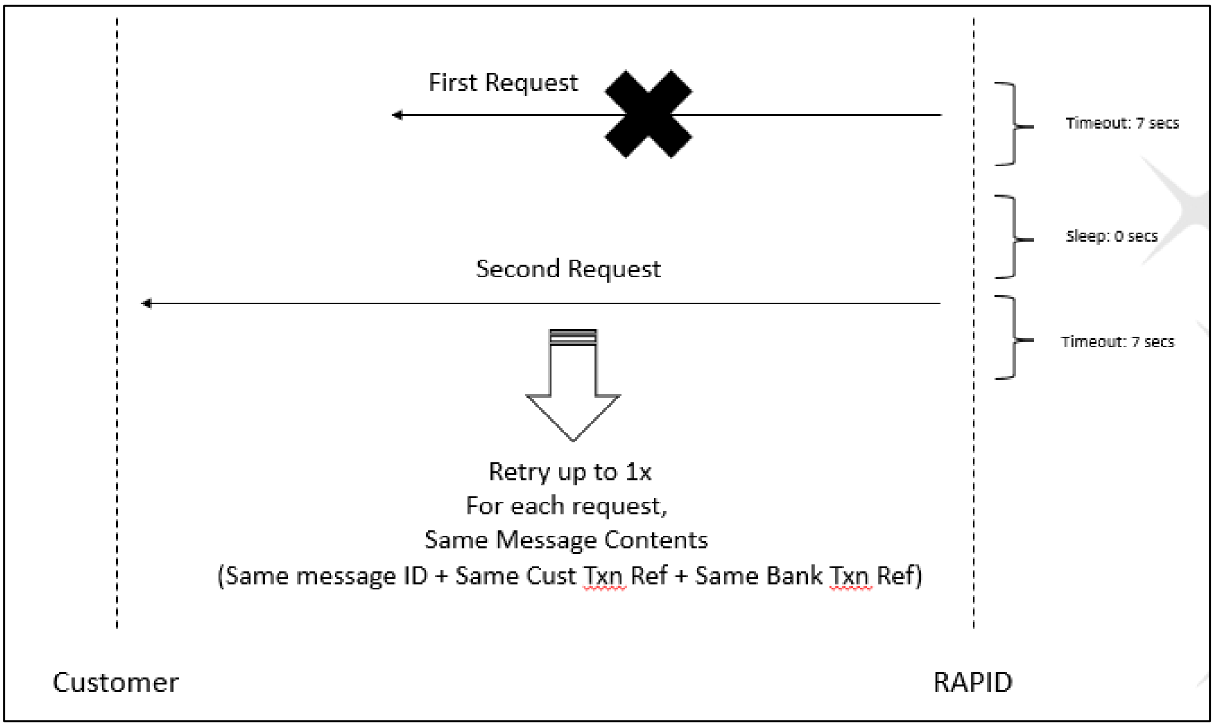
i) Sync API

Customer can retry up to 2 times with the same customer transaction reference but different message id if getting timeout for sync API request to fetch the latest transaction status.



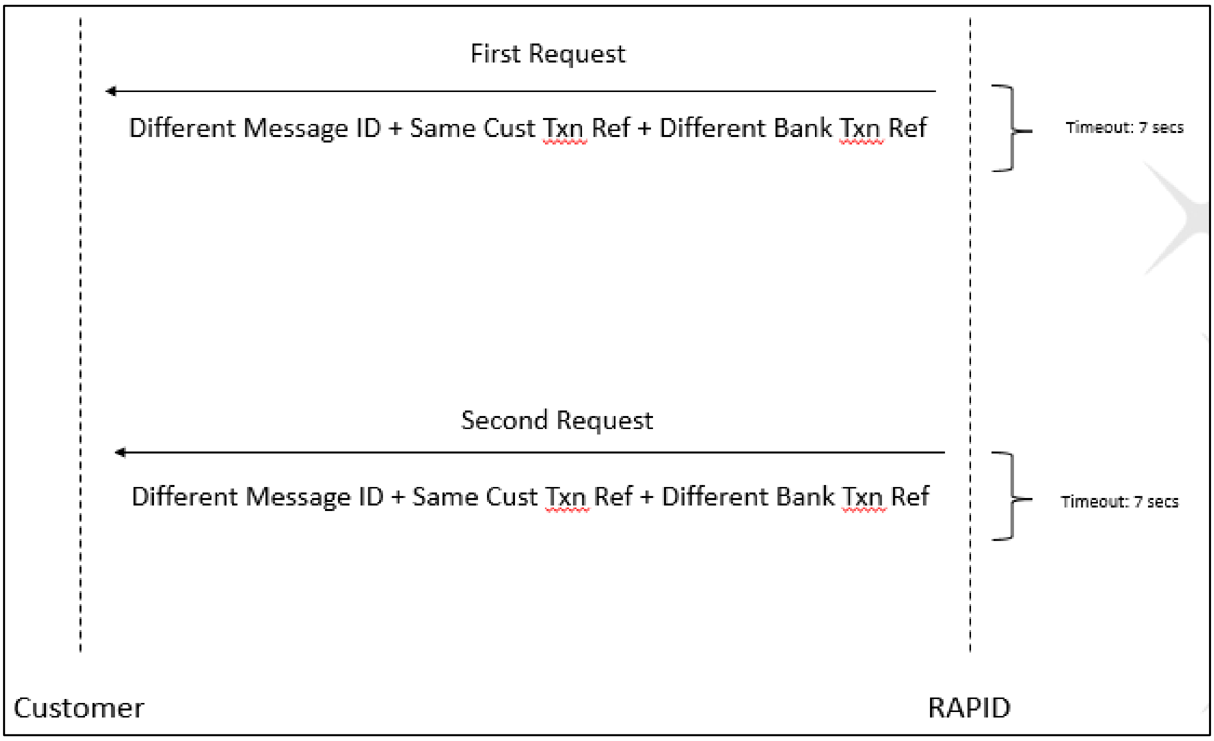
ii) **Async API - DBS Retries**

In the event that the bank did not receive a http 200 response for the ICN API message that it sent to customer's server, the bank will retry once and fire the same ICN API message to customer's server again. Customers should implement a retry handling logic to detect same bank transaction reference and/or message ID within ICN API messages received, and ignore the subsequent ICN API message.



iii) Async API - Consumer Scan / Pay Multiple Times

When consumers scan and pay the same QR more than one time, corporate customers may also receive duplicate ICNs. These ICNs would contain the same customer transaction reference, but have different bank transaction reference. The different bank transaction references represent the multiple credits that have occurred on the corporate customer's account (i.e. consumers pay more than once to the corporate customer's account).


First Request
Different Message ID + Same Cust Txn Ref + Different Bank Txn Ref
Timeout: 7 secs
Second Request
Different Message ID + Same Cust Txn Ref + Different Bank Txn Ref
Timeout: 7 secs
Customer
RAPID

4.3 Trade APIs

Trade customers would be assigned an unique ORG-ID with account number for RAPID API subscriptions. Each ORG-ID is unique and must correspond to a legal entity registered with the local regulatory body.

For all trade documents (supporting) sent by customer to the bank, the customer should follow the following file naming convention.

DocumentName: File Naming Convention

<OrgID><TradeProductType><DocumentName><DDMMYYHHMMSS> . <doctypeExtension>

Where

- *orgID = the unique identifier assigned by DBS to a legal corporate entity*
- *tradeProductType =*

- EDC
- APF
- ARF
- LC
- IBLC

- *documentName must be perpertually unique,*

- *DDMMYYHHMMSS must be the date / time the document was uploaded to DBS, and the*

- *doctypeExtension must reflect the actual document type of the attachment.*

- *Documents must not be more than 5mb in size.*